



Valid BST

A binary tree is a multi-node data structure where each node has, at most, two child nodes and one stored value. It may either be:

1. An empty tree, where root = null, or
2. A tree with a non-null root node that contains a value and two sub-trees, *left* and *right*, which are also binary trees.

A binary tree is a binary search tree (BST) if all the non-null nodes exhibit two properties:

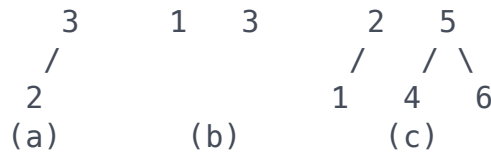
A binary tree is a binary search tree (BST) if all the non-null nodes exhibit two properties:

1. Each node's left sub-tree contains only values that are lower than its own stored value.
2. Each node's right sub-tree contains only values that are higher than its own stored value.

Pre-order traversal is a tree traversal method where the current node is visited first, then the left sub-tree and then the right sub-tree. If the left subtree has children, they are traversed in the same order, left then right, before moving on to the original right subtree. The following pseudo code parses a tree into a list using pre-order traversal :

1. If the root is null, output the null list.
2. For a non-null node:
 1. Make a list **L** by pre-order traversing the left sub-tree.
 2. Make a list **R** by pre-order traversing the right sub-tree.
 3. Output the stored value of the non-null node, append **L** to it, then append **R** to the result.





For the three trees above, the pre-order traversal list will be:

- (a) 1 3 2
- (b) 2 1 3
- (c) 3 2 1 5 4 6

Given a list of numbers, determine whether they can represent the pre-order traversal list of a binary search tree (BST).

Input

The first line contains the number of test cases, **T**. The next **T** test cases follow, consisting of two lines each.

The first line of each test case contains the number of nodes in the tree, **N**. In next line there will a list of **N** unique numbers, where each number is from the range $[1, N]$.

Output

For each test case, print the string "YES" if a BST, whose pre-order traversal is equal to the list, exists. Otherwise print the string "NO". Do not print quotes. Preserve capitalization.

Constraints

$$1 \leq T \leq 10$$

$$1 \leq N \leq 100$$

Sample Input

```
5
3
1 2 3
3
2 1 3
6
3 2 1 5 4 6
4
1 3 4 2
```

```
5
3 4 5 1 2
```

Sample Output

```
YES
YES
YES
NO
NO
```

Explanation

- The first three cases are from the above examples.
- In case 4, after encountering the 3, the 4 tells us we are on the right subtree, which means that no values smaller than 3 are allowed. So when we see the 2 we know the list is invalid.
- In case 5, after encountering the 3, the 4 and 5 tell us we are on the right sub-tree, so the subsequent encounter of values 2 and 1, which belong in the left sub-tree, tells us that the list is not valid as a pre-order traversal of a BST.

YOUR ANSWER

Java 7

[Click here](#) to know how to read from STDIN and write to STDOUT

```
1 import java.io.*;
2 public class Solution {
3     public static void main(String args[] ) throws
Exception {
4         /* Enter your code here. Read input from STDIN.
Print output to STDOUT */
5     }
6 }
```

Line: 1 Col: 1

☐ Test against custom inputRun CodeSubmit code & Continue

 [Download sample testcases](#) The input/output files have Unix line endings. Do not use Notepad to edit them on windows.

[About](#) [Privacy policy](#) [Terms of service](#)